In the claims:

For the Examiner's convenience, all pending claims are presented below with changes shown in accordance with the mandatory amendment format.

1. (Currently Amended) A method comprising:

at a start of a block of code, performing only one test for the block of code to determine if resources of an architectural stack are available for the block of code, the block of code including multiple instructions adding data to the stack or removing data from the stack; and

signaling an error if said resources of the architectural stack needed for said block of code are not available.

2. (Original) The method as claimed in claim 1, said method further comprising:

determining a set of available resources that will be available after said block of code has executed.

3. (Canceled)

4. (Previously Presented) The method as claimed in claim 1 wherein the availability of the stack resources are determined at a compile time.

5. (Previously Presented) The method as claimed in claim 1 wherein the availability of the stack resources are determined dynamically.

6. (Currently Amended) The method as claimed in claim 1 wherein signaling said error if said resources of the architectural stack needed for said block of code are not available comprises branching to a fault handler routine.

7. (Original) The method as claimed in claim 6 wherein signaling said fault handler routine simulates a processor exception.

8. (Previously Presented) The method as claimed in claim 1 wherein the stack resources are represented by a bit vector.

9. (Original) The method as claimed in claim 8 wherein said bit vector is generated dynamically.

10. (Currently Amended) A computer-readable medium having stored thereon a set of instructions to monitor processor resources, said set of instruction, which when executed by a processor, cause said processor to perform a method comprising:

at a start of a block of code, performing only one test for the block of code to determine if resources of an architectural stack are available for the block of code, the block of code including multiple instructions adding data to the stack or removing data from the stack; and

signaling an error if said resources of the architectural stack needed for said block of code are not available.

11. (Original) The computer-readable medium as claimed in claim 10, wherein said set of instructions further includes additional instructions, which when executed by said processor, cause said processor to perform said method further comprising:

determining a set of available resources that will be available after said block of code has executed.

12. (Canceled)

13. (Previously Presented) The computer-readable medium as claimed in claim 10 wherein the availability of the stack resources are determined at a compile time.

14. (Previously Presented) The computer-readable medium as claimed in claim 10 wherein the availability of the stack resources are determined dynamically.

15. (Currently Amended) The computer-readable medium as claimed in claim 10 wherein signaling said error if said resources of the architectural stack needed for said block of code are not available comprises branching to a fault handler routine.

16. (Original) The computer-readable medium as claimed in claim 15 wherein signaling said fault handler routine simulates a processor exception.

17. (Previously Presented) The computer-readable medium as claimed in claim 10 wherein the stack resources are represented by a bit vector.

18. (Original) The computer-readable medium as claimed in claim 17 wherein said bit vector is generated dynamically.

19. (Currently Amended) A computer-readable medium, having stored thereon a first set of instructions, the first set of instructions, which when executed by a processor, generate a second set of instructions through a binary translation process, the second set of instructions when executed by the processor, cause said processor to perform a method comprising:

at a start of a block of code, performing only one test for the block of code to determine if resources of an architectural stack are available for the block of code, the block of code including multiple instructions adding data to the stack or removing data from the stack; and

signaling an error if said resources of the architectural stack needed for said block of code are not available.

20. (Original) The computer-readable medium as claimed in claim 19, wherein said set of instructions further includes additional instructions, which when executed by said processor, cause said processor to perform said method further comprising:

determining a set of available resources that will be available after said block of code has executed.

21. (Canceled)

22. (Previously Presented) The computer-readable medium as claimed in claim 19 wherein the availability of the stack resources are determined dynamically.

23. (Currently Amended) The computer-readable medium as claimed in claim 19 wherein signaling said error if said resources of the architectural stack needed for said block of code are not available comprises branching to a fault handler routine.

24. (Original) The computer-readable medium as claimed in claim 23 wherein signaling said fault handler routine simulates a processor exception.

25. (Original) The computer-readable medium as claimed in claim 19 wherein needed resources are represented by a bit vector.